DOCUMENT RESUME

ED 268 979 IR 012 042

AUTHOR Hart, Roland J.; Goehring, Dwight J.
TITLE Computer-Assisted Scheduling of Army Unit Training: An Application of Simulated Annealing.
PUB DATE Aug 85
NOTE 28p.; Paper presented at the Annual Convention of the American Psychological Association (Los Angeles, CA, August 23-27, 1985).
PUB TYPE Reports - Evaluative/Feasibility (142) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Armed Forces; *Computer Simulation; Costs; Heuristics; *Military Training; Problem Solving; *Scheduling; Systems Approach; Time Management
IDENTIFIERS *Army

ABSTRACT

This report of an ongoing research project intended to provide computer assistance to Army units for the scheduling of training focuses on the feasibility of simulated annealing, a heuristic approach for solving scheduling problems. Following an executive summary and brief introduction, the document is divided into three sections. First, the Army training scheduling problem is described, with references to specific needs for different types of schedules for different echelons, types of scheduling constraints, time availability, downward inheritance of events and activities prescribed by higher echelons, command priorities, intra- and inter-schedule compatibility, and input-output requirements for scheduling. Simulated annealing is next introduced as a solution for scheduling problems and its process is described. An application of this algorithm is illustrated via an Army test program written in FORTRAN 77 on a VAX 11/780 computer, which focused on scheduling at the battalion level. A description of the test program (with flow diagram), methods of its evaluation, and a sample run of training schedules for five companies are provided. The third section presents conclusions and recommendations emerging from this application of the test program. Four figures, one table, and a list of references are provided. (JB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED268979

U.S. DEPARTMENT OF EDUCATION
OERI
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)
X This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official position or policy.

COMPUTER-ASSISTED SCHEDULING OF ARMY UNIT TRAINING: AN APPLICATION OF SIMULATED ANNEALING

Roland J. Hart
Dwight J. Goehring

Army Research Institute
Presidio of Monterey, California

IR012042

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Roland Hart

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

CONTENTS

Page

INTRODUCTION..... 1

DESCRIPTION OF ARMY TRAINING SCHEDULING PROBLEM..... 2

Types of Schedules for Different Echelons..... 2

Types of Scheduling Constraints..... 4

Input/Output Requirements for Scheduling..... 6

USE OF SIMULATED ANNEALING TO SOLVE THE SCHEDULING PROGRAM..... 8

Description of Simulated Annealing..... 8

Illustrative Application of Simulated Annealing.....10

CONCLUSIONS AND RECOMMENDATIONS.....19

REFERENCES.....21

APPENDIXES

A. Computer Program Listing.....A-1

B. Sample Program Run with Data Tables.....B-1

LIST OF TABLES

1. Illustration of Schedular Functioning for Monday, Company B (from Appendix B)

LIST OF FIGURES

1. The Relationships Between Echelons and Schedules for ITMS System

2. Data Structures and Flow of Control for Simulated Annealing Test Program

3. Operation of the Simulated Annealing Scheduling Algorithm

4. Relative Cost by Iteration for Best Schedule Sets Using Easy and Difficult Scheduling Problems

EXECUTIVE SUMMARY

Requirement:

In Army units, numerous required training and non-training activities compete for time. The scheduling of training in the Army is complex when done manually, but computer assistance can reduce the complexity of training scheduling for planners and lead to better planning of training. Improved training planning can make more efficient use of time and resources, and thus contribute to improving readiness.

Our objective is to provide a realistic description of the Army training scheduling problem and to explore the feasibility of a particular heuristic approach for solving this problem. The approach taken is called simulated annealing.

Procedure:

The Army training scheduling problem is complex because it occurs across different echelons and involves three different types of calendars/ schedules. Priorities of higher echelons almost always predominate over those of lower echelons. In addition, many constraints exist for the Army training scheduling problem. Training and non-training tasks require time and resources. Resources can be either reusable (e.g., ranges) or expendable (e.g., ammunition). Further, activities may have temporal constraints, such as requiring some tasks to be trained ahead of others, or co-occurrence constraints requiring some tasks to be trained simultaneously with others. There are also command priority constraints at each echelon which affect scheduling requirements. In addition, the priority given training varies at different times. To provide adequate computer assisted training scheduling, the full diversity of the Army's scheduling problem must be accommodated.

Simulated annealing was selected as a method with high potential for dealing with both the complexity and heterogeneity of the problem. Simulated annealing, is appropriate when, as with scheduling, multiple "good" solutions are acceptable as opposed to an uniquely "best" or optimal solution. The implementing algorithm operates iteratively by: (1) random assignment of activities to times, (2) extensive modification due to constraint accommodations, and (3) evaluation of the schedule set using a cost function. The constraint accommodation proceeds by satisfying the most important constraints first followed by the less important constraints. The cost value is used to determine the extent to which improvement in schedule set quality has been achieved.

Findings:

A scheduling system was written as a test program designed to permit evaluation of the feasibility of simulated annealing for solving the Army training scheduling problem. The system was successful in demonstrating that this approach is a promising one, warranting further research. Several augmentations of the current implementation of the simulated annealing approach are suggested.

Utilization of Findings:

It appears highly likely that an applications version of a simulated annealing system can be developed and implemented on the Integrated Training Management System (ITMS), an automated unit management system at Fort Lewis, Washington. User friendliness and adaptable reports generation for an applications scheduling system will be facilitated by the use of a relational Data Base Management System (DBMS).

COMPUTER-ASSISTED SCHEDULING OF ARMY UNIT TRAINING: AN APPLICATION OF SIMULATED ANNEALING

This report describes research that is part of an ongoing project ultimately intended to provide computer assistance to Army units for the scheduling of training. The objective here is to present a realistic description of the Army training scheduling problem and to evaluate the feasibility of a particular heuristic approach to the problem called simulated annealing.

The Army scheduling problem is characterized by competition between training and non-training tasks for scarce time and material resources. The large number of training and non-training tasks and their associated time and resource requirements make scheduling extremely difficult. The scheduling problem is further complicated by the following facts: (a) different echelons are involved in creating the schedules and calendars, (b) different calendars are used to plan for different lengths of time into the future, (c) a variety of scarce resources are required for training, and different echelons may control different resources, and finally (d) a wide variety of different types of scheduling constraints are involved, including requirements for the inclusion of command priorities and the inclusion of some activities as prerequisite for others.

Computer automated assistance for scheduling training offers a potential for improving the allocation of training time and resources in addition to saving time of planners responsible for accomplishing the scheduling task. Better allocation of training time to training tasks and more efficient use of scarce training resources are expected to yield increased unit readiness.

An additional advantage of computer automated assistance is its potential to help planners with comparison planning. Comparison planning involves the ability to ask "what if" questions, and then compare the resultant schedules based on alternate options. For example, higher echelons may not fully understand the impact on lower echelon schedules of schedule changes that are made at higher levels. However, computer automated assistance would provide the opportunity for higher echelons to understand the relative costs of making possible changes prior to actually making them. Thus, the impact of schedule changes can be identified and negative effects minimized.

The scheduling system described here is designed to lead to a system to be implemented on the Integrated Training Management System (ITMS), at Fort Lewis, Washington (see Army Development and Employment Agency [ADEA], January, 1985). The ITMS is an experimental prototype system demonstrating computer automation of unit training management. A system like the prototype may eventually be implemented widely throughout the Army. The Army training scheduling problem exists in its full complexity on the ITMS.

The scheduling system that is described here builds upon previous work on automated training scheduling and resource allocation, sponsored by the Army Training Board (Van der Eijk, Ignizio, & Yang, 1981; Yang & Ignizio, 1982; Medeiros & Yang, 1983; Medeiros & Yang, 1983b). This work includes a computer program designed to schedule training during mobilization called The Mobilization Resources Scheduling Program. This previous work was not

designed specifically for implementation on ITMS and for this reason does not address the Army training scheduling problem in its full complexity as a multi-schedule, multi-echelon problem. Further, the previous efforts address many but not all of the training scheduling constraints important in a practical system.

The following discussion is organized into three sections. First, the Army training scheduling problem is described. Second, the use of simulated annealing as a solution for the scheduling problem is illustrated. The third section presents conclusions and recommendations emerging from the application of this approach. The computer program listing and a sample program run with data tables can be found as appendices to the report.

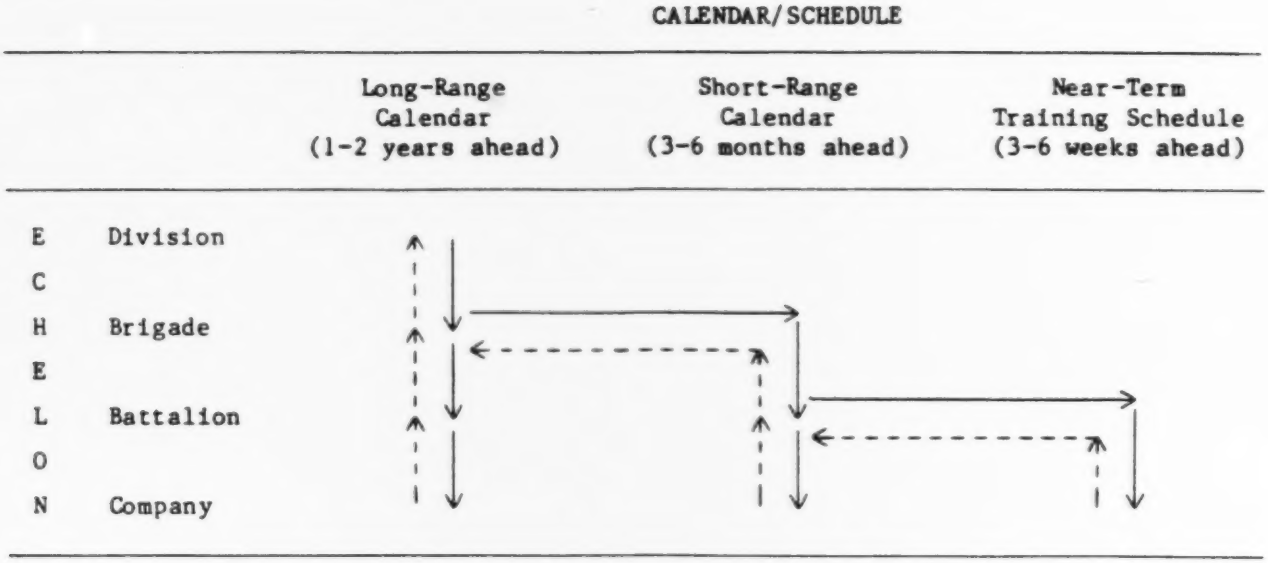
DESCRIPTION OF ARMY TRAINING SCHEDULING PROBLEM

Types of Schedules for Different Echelons

The training scheduling problem is defined here for all echelons within an Army division (see ADEA, 1985). The echelons within an Army division can be ordered from smallest to largest: company (about 150 soldiers); battalion (about 750 soldiers); brigade (about 3,000 soldiers); division (about 15,000 soldiers). There are three different training calendars/schedules that differ primarily in how far into the future needs are projected. These training calendars/schedules have been denoted "Long-Range Calendar" with projections up to 24 months in the future, "Short-Range Calendar" with projections from three to six months in the future, and "Short-Term Training Schedules" that extend from three to six weeks in the future. Different echelons have primary responsibility for preparing and updating the three types of calendars/schedules. The relationships between echelons and the three types of calendars/schedules are shown in Figure 1.

The strict hierarchical organizational structure of the Army gives a top-down character to the process of training scheduling. Higher echelons may prescribe specific training or other activities. Major training events that originate at the division or higher level are first placed on the Long-Range Calendar. The predominant time units employed in the Long-Range Calendar are weeks, but specifications of days, such as holidays, also appear. Division has primary responsibility for preparation and revision of the Long-Range Calendar, although subordinate brigades and battalions may also be consulted. Typically the interest of the brigades and battalions is limited to those parts of the overall long-range calendar which concern them. Division has the responsibility for overall consistency and resolution of conflicts between subordinate units.

The Short-Range Calendar is prepared primarily at the brigade and battalion echelons. The predominant unit of time employed is days. The Short-Range Calendar takes mandated activities and events from the relevant time-frame of the Long-Range Calendar and adds them to the Short-Range Calendar, but with increased specificity and detail. Short-Range Calendars are coordinated with Division. However, Short-Range Calendars are ultimately subordinate to decisions and event scheduling occurring at higher echelons. Planning for the allocation of training resources like ranges, ammunition, MILES equipment, and training devices is accomplished on Long- and Short-Range Calendars.



- - - -> = recommendation

——> = decision

horizontal lines indicate transfers (or recommended transfers) of events between calendars/schedules

Figure 1. The Relationship Between Echelons and Schedules for ITMS System

Near-term Training Schedules are prepared and maintained at the company level. Training schedules depict, on a weekly basis, specific training activities in great detail. The predominant unit of time employed goes down to hours and even minutes. Many activities on Training Schedules are derived from Short-Range Calendars. Training Schedules are posted for the purpose of providing information to the members of the company, so that they can perform the activities. For this reason Training Schedules contain specific information not found on Short-Range Calendars. The detailed information typically includes precise times, locations, trainers and trainees involved, type of uniform, references, and remarks. The Training Schedule for each company is coordinated at the level of the parent battalion. Possible inconsistencies within a schedule are identified and conflicts between company-level Training Schedules are resolved. A typical conflict might involve two companies that request the same battalion classroom for the same time period.

The overall scheduling task is typified by the downward inheritance of events and activities prescribed by higher organizational echelons, coupled with additions, augmentation, and greater specificity at each successively lower level. There is an iterative process of allocating time to activities. Conflicts in allocation are resolved at the echelon just above the conflict. Activities scheduled vary in terms of command priority, resource requirements, relationships to other activities, and size. The size of scheduled activities may vary from a brigade level field training exercise to individual level skill training on specific tasks like first aid.

One practical aspect of the Army training scheduling problem that is important to appreciate is the necessity for frequent rescheduling. For a variety of reasons, activities dictated by higher echelons often preempt activities scheduled on company-level Training Schedules or unanticipated circumstances disrupt Training Schedules, even after the company level activities have been approved and considered final. Thus, at the company level rescheduling is common and requires decisions regarding which company-level training activities to reschedule as soon as possible, which ones to postpone, and which ones to eliminate. Assistance with rescheduling company-level training activities is a necessary practical feature for an automated scheduling system. When carrying out the rescheduling of activities, an automated system should minimize, as much as possible changes to existing calendars and schedules.

Types of Scheduling Constraints

There are distinct classes of scheduling constraints that shape Army training calendars and schedules. These contraints influence scheduling decisions differently and vary in their relative priority for being met. Satisfaction of relevant constraints serves as a useful criterion for judging the quality of training calendars. That is, relatively "good" schedules violate few scheduling constraints, while relatively "poor" schedules violate many contraints. Thus, constraint violations can be used in an automated

system to construct a "cost function"[1] for judging the relative quality of alternative calendars generated by the computer. These constraints, described below, relate to: time availability, downward inheritance, command priorities, inter-schedule compatibility and intra-schedule compatibility.

Time availability. The most basic factor affecting scheduling is time available for training. A sufficient amount of time to accommodate all of the training that ideally needs to be performed simply does not exist. This means that some desirable training activities must, of necessity, be left out of the Training Schedule thereby reducing the quality or increasing the cost of a schedule. Further, training activities vary in the amount of time they require for completion. Longer activities are often viewed as more important than shorter activities. Thus, other factors remaining constant, those activities using more time can be considered "more expensive" if they cannot be fit into the schedule. Therefore, if it becomes impossible to include these longer tasks in the Training Schedule, their omission will add more to the cost function than the omission of shorter duration activities.

Downward inheritance. Activities or events are often fixed at specified times in the calendars and schedules by higher echelons and inherited downward. Thus, lower echelons inherit tasks or events specified by higher echelons. Such activities may be assigned a specific time or not. If times are specified, then lower level calendars and schedules must maintain the times established by higher echelons. Otherwise the lower echelons can assign the activities to times. Activity times may also be mandated in terms of fixed blocks of time. Subsequent scheduling of activities at a given echelon must be accomplished without alteration of activities that have been inherited from superordinate echelons. In addition, activities can be fixed at the current echelon. This means that all activities must be scheduled without altering these activities fixed at the current echelon as well as additional activities that may have been inherited from above. When conflicts arise that cannot be resolved, and activities remain unscheduled, cost is incurred.

Command priorities. The training priorities of commanders or their designated representatives at each organizational level must differ and be taken into account in the development of calendars and schedules. In the case of conflict in priorities for special training activities, the command priorities of higher echelons predominate over priorities of commanders at subordinate echelons, although communication between echelons may be possible to resolve differences. Under some conditions, particular units may be given higher priorities than other units for all training activities and resources. This situation might occur, for example, when some units must be prepared to mobilize more quickly than others.

Inter-schedule compatibility. Since company-level units each have their own Training Schedules, conflicts can arise between Training Schedules for different units. The most frequent source of conflicts arises from

---

[1]A cost function produces numbers that canbe used to judge the relative quality of schedules. These numbers should not be taken literally as budget costs in dollars and cents for accomplishing training.

requirements for scarce training resources. Resource requirements fall into two categories: expendable resources such as ammunition and gasoline, and renewable resources such as training facilities and equipment. Expendable resources are diminished in amount as a result of use, while renewable resources are not. Conflicts for expendable resources arise from shortages in supply. Eliminating conflict requires increasing supplies. Conflicts for renewable resources arise when demand at a particular time exceeds availability. Thus, renewable resources may be scarce at one time but not another depending upon the timing of demands. In this case, it is possible to eliminate the conflict by reassigning the times given to competing activities in different Training Schedules.

Intra-schedule compatibility. Conflicts can arise in the scheduling of events within a unit's Training Schedule. These conflicts may arise when there are: 1) temporal relationships between events, 2) temporal requirements for specific events, or 3) restrictions on the use of particular time blocks for training. A frequent temporal relationship between events occurs when training activities serve as prerequisites for other activities and must be trained ahead of the other activities yielding a precedence relationship. Violating such temporal relationships between events creates conflict within a Training Schedule. In other cases, specific training activities must occur together, producing synchronous or co-occurrence constraints for scheduled activities. Other temporal relationships between events are also possible, such as activities that must occur contiguously. Temporal requirements for specific events occur when activities have particular time-related requirements, such as darkness for training of night vision devices. The above temporal constraints may be established by commanders or pre-established in a data base. The data base may use doctrine, policy, and logical necessity as criteria for temporal constraint relationships.

Restrictions on time blocks for training generally occur at division level. Time is often blocked into groups of weeks labeled red, green, and yellow time that indicate the priority of training activities for subordinate units during a given block of time. In green time, training activities and events have priority for a unit, while in red time garrison support activities have priority. During yellow time (when this block exists) training activities have an intermediate priority. Intra-schedule conflicts are created when restrictions on the use of blocks of time are violated.

Input-Output Requirements for Scheduling

An automated system for scheduling can be divided into two distinct parts: An algorithmic and a human interface component. The algorithmic component employs an optimization methodology designed to minimize the cost function. Thus, the algorithmic part must be based on an appropriate problem definition and cost function. The human interface component involves program input and output.

While the input and output requirements are vital for any operational implementation of an automated scheduling system, the algorithmic aspects are a natural prerequisite. Thus, the primary focus of this paper is on the algorithmic aspect of the problem, including definition of the problem and description of the algorithm. The general feasibility of the approach is tested using an initial implementation of algorithm concepts. However, a

brief discussion of requirements for input and output is presented first in order to provide an operational view of the applications system.

The primary requirement for input to and output of a scheduling system is that it be "user friendly". In order to create a practical system that will be used within Army units, it is important to keep program commands and data entry as simple and easy to learn as possible. In addition, the scheduling system should generate output information including summary reports, tailored to specific users' needs. In order to facilitate user friendliness and generate useful summary reports, the input-output component of the scheduling system will employ a relational data base management system (DBMS). A relational DBMS system is designed to minimize input-output requirements for program users. For example, in a relational DBMS language, users are not required to keep track of the details of where variables are stored.

Input to the automated scheduling system will be facilitated by a menu-driven database sub-system containing tables or lists of training activities and events. These lists of training activities can be linked to lists of resource requirements for training activities. These lists will reduce data entry requirements by allowing users to select items from a list in lieu of entering new data. Ideally, the system will also generate reminders to program operators about those training activities that should be periodically trained. These reminders may be generated from an adjunct system that records or predicts training proficiency over time.

Another important input-output feature of an automated scheduling system is feedback to users regarding scheduling failures. It will often be impossible to produce a set of Training Schedules that completely accommodates a highly constrained problem. The user needs to be informed of scheduling failures and the reasons for each failure so that appropriate action can be taken. For example, it may be impossible to schedule a Field Training Exercise due to the shortage of a renewable resource like MILES equipment. This shortage is potentially solvable by reallocating the timing of events. On the other hand, the shortage of an expendable resource like fuel is not solvable unless additional expendable resources are obtained. If users are informed of the reasons that events remained unscheduled, they may be able to take appropriate action to improve schedules (e.g., increase resources, or modify priorities).

The most basic reports required by an automated scheduling system are the Long- and Short-Range Calendars and Training Schedules. Updated Calendars and Training Schedules should be produced and distributed as modifications are made and approved. Additional reports that highlight changes in scheduled activities and changes in resource requirements may be particularly helpful. For example, if a renewable resource (e.g., classroom) is no longer required by one unit at a particular time, a report could highlight this fact, noting new availabilities of renewable resources. Reports related to resource requests (e.g. ammunition) and resource utilization could also be produced (see ADEA, 1985). For renewable resources, it may be helpful for report resource usage as a function of unit and time.

Of particular interest at higher echelons in the Army is the issue of training cost. Reports can be extended to cover training cost as a function of resource utilization and resource cost. Such reports would require

appropriate "cost models" that take into account the type of unit, the unit training location and additional relevant factors. Such training cost models would be based, at least in part, on Training Calendars and Schedules. For this reason, the automated scheduling work reported here should contribute to the automation of reports related to training cost.

USE OF SIMULATED ANNEALING TO SOLVE THE SCHEDULING PROBLEM

Description of Simulated Annealing

An optimization methodology is needed to solve the Army training schedule problem. Simulated annealing (Kirkpatrick, Gelatt & Vecchi, 1983) was selected as an appropriate method because of the match between the capabilities of the methodology and the characteristics of the scheduling problem. First, it can handle large problems. The training scheduling problem for an entire Army division is large. In addition, it can handle virtually any type of constraint. This feature is important because the Army training scheduling problem includes a wide variety of constraints. The approach is flexible in the sense that constraints can be changed or added and the method can still work. Flexibility is important because frequent modification is expected in an operational environment.

Another feature of simulated annealing that matches the Army training scheduling problem is size of the solution space. Application of simulated annealing is appropriate when multiple "good" solutions are acceptable as opposed to a uniquely "best" solution. The training scheduling problem matches this description. There is a relatively large class of "good" schedules and the goal is to find one of the schedules in this class. The solution space can be visualized geometrically as ridges and valleys, with multiple good solutions found along the valleys. Using this analogy increasing height corresponds to increasing cost or disutility. The "goal" is to not necessarily find the lowest point but a solution in a valley. The "best" possible solution is not likely to be much better than other solutions in the "good" class.

Further, for large combinational problems like the scheduling problem, it becomes impossible to check all possible combinations, in search of the one combination that minimizes the cost function, because of time requirements. The time requirements for such problems increase exponentially with N (i.e., number of activities to be scheduled). Rather than search all combinations, optimization methods use heuristics to minimize cost functions. Such heuristics can be generally classified as either "divide-and-conquer" or iterative improvement strategies (Kirkpatrick, Gelatt & Becchi, 1983). "Divide-and-conquer" or decomposition strategies partition a large problem into smaller ones that can be solved separately. The separate solutions then must be recomposed together to provide an overall solution. This approach is most appropriate when sub-problems are naturally disjoint so that the sub-problem solutions can be generated and then combined to form an overall solution without excessive distortion. For iterative improvement, a standard rearrangement operation is applied until a combination is found that improves the cost function. Typically, the rearranged configuration then becomes the new configuration. The standard rearrangement operation is repeated until no further improvement in the cost function can be found. This type of search may get stuck in a local but not global minimum. For this reason, the

iterative improvement search is often repeated using different random starting points. The simulated annealing method contains characteristics of both "divide-and-conquer" and iterative improvement strategies.

Finally, the simulated annealing algorithm does not require a set computer time to obtain a solution. The solution will get better on the average the longer the program runs. A computer run can be terminated when a point of diminishing returns is achieved.

The simulated annealing heuristic operates by analogy to annealing in physical systems. Annealing in a physical system involves repeated heating and cooling of liquids or solids. When a liquid is hot there is a rapid random movement of the molecules making up the liquid. Heating a liquid increases the rapidity of the random movement of molecules. By contrast, cooling a liquid slows the random movement of molecules until they reach the point where they are frozen in place. Molecules frozen in place are considered "cold."

As applied to a scheduling system, the molecules are activities or events that must be scheduled. Random movement involves the random assignment of activities to times. The concept of temperature is linked to the cost function, and refers to the degree of random movement permitted relative to the cost function. High temperature is associated with high values of the cost function, low temperature is associated with low values of the cost function. High temperature means that random assignment of activities to times is permitted no matter how large the cost function gets as a result of the assignments. That is, assignments are not constrained by the cost function. On the other hand, cold temperature means that only some random assignments of activities to times are permitted, namely, assignments that reduce or at least do not increase the cost function.

The operation of the simulated annealing heuristic involves successively "heating" and "cooling" the system in search of a "good" schedule defined by a low cost function. The process of successively "heating" and "cooling" the system is accomplished to avoid the problem of getting stuck in local minimums. The problem of getting stuck in local minimums is a common one for problems based on iterative improvement (e.g., greedy algorithms). The simulated annealing heuristic requires the creation of an appropriate "annealing schedule" involving the extent and frequency of heating and cooling. An advantage of the simulated annealing heuristic is that annealing schedules help overcome the local minimums problem.

In addition to creating an appropriate annealing schedule, an algorithm is necessary to cool the system. (Creating a hot system is not difficult since heating simply entails unconstrained random assignment of activities to times.). Cooling the system, however, requires development of an algorithm, created especially for the problem at hand, that constrains the random assignments permitted to those producing successively lower values of the cost function, as the system is cooled. In order to "cool" a scheduling system, those tasks that have the greatest potential for increasing the cost function are scheduled first before the schedule gets too full to include them. Then tasks that have less potential for increasing the cost function are successively scheduled. Unconstrained tasks that minimally add to the cost function are scheduled last. This strategy increases the probability that

important activities and constraints are accommodated in the resultant schedules. The above ordered scheduling builds "divide-and-conquer" strategy into the cooling algorithm. That is, the overall problem of scheduling is partitioned into a series of smaller sub-problems requiring scheduling subsets of activities in order of their importance.

Illustrative Application of Simulated Annealing

A simulated annealing heuristic approach to the Army training scheduling problem was implemented using a test program to explore its feasibility. Feasibility was examined using two criteria: computer efficiency (e.g., time requirements, memory requirements), and the effectiveness of program operations. The effectiveness of program operation includes the face validity of the training schedules that are produced, and the flexibility of the program to handle additional requirements. The program described here is intended as an initial feasibility test. Assuming that the approach appears feasible, the test program can then be used to "fine tune" the design of the heuristic of the selection and operation of: (a) the cost function, (b) annealing schedules, and (c) types of constraints.

Description of the Test Program. The Army scheduling problem was delimited by focusing on scheduling at the battalion level. This level was selected because it contains scheduling features from all echelons. If the annealing heuristic is effective at this level, one can assume that it can be applied at all levels with the appropriate modifications for each echelon.

All five companies within a single battalion are scheduled simultaneously in the test program. These five companies inherit training activities and events from the Long- and Short-Range Calendar created at higher echelons. The test program uses one-hour units of time for activities and forty-hour weeks. A final program will require smaller time units and the potential for longer weeks. The initial test program has only one annealing schedule for "cooling" the system. This schedule flows from the top to bottom following the activity importance criterion.

The test program employs all of the important categories of scheduling constraints: (a) inheritance of activities from higher echelons and "fixing" activity times, (b) interschedule conflicts of both renewable and expendable resources, (c) company-level training activity priorities, (d) different unit priorities assigned to different companies, and (e) intra-schedule conflicts based on of temporal constraints for activities (i.e., before/after, immediately before/immediately after, and temporal ordering of sequences of contiguous activities). The constraints that were considered most important were included in the test program. Any exclusions were simply due to limitations on time and resources, and can be added as required for an applications version of the program.

The simulated annealing test program was written in FORTRAN 77 on a VAX 11/780 computer. The FORTRAN code for the test program is provided at Appendix A. An overview of the data structures and flow of control of the simulated annealing test program is shown in Figure 2. Blocks A through E in Figure 2 depict five data structures containing the necessary input information. An example of data input files and output schedules for an example problem is provided in Appendix B and is discussed later in this section.

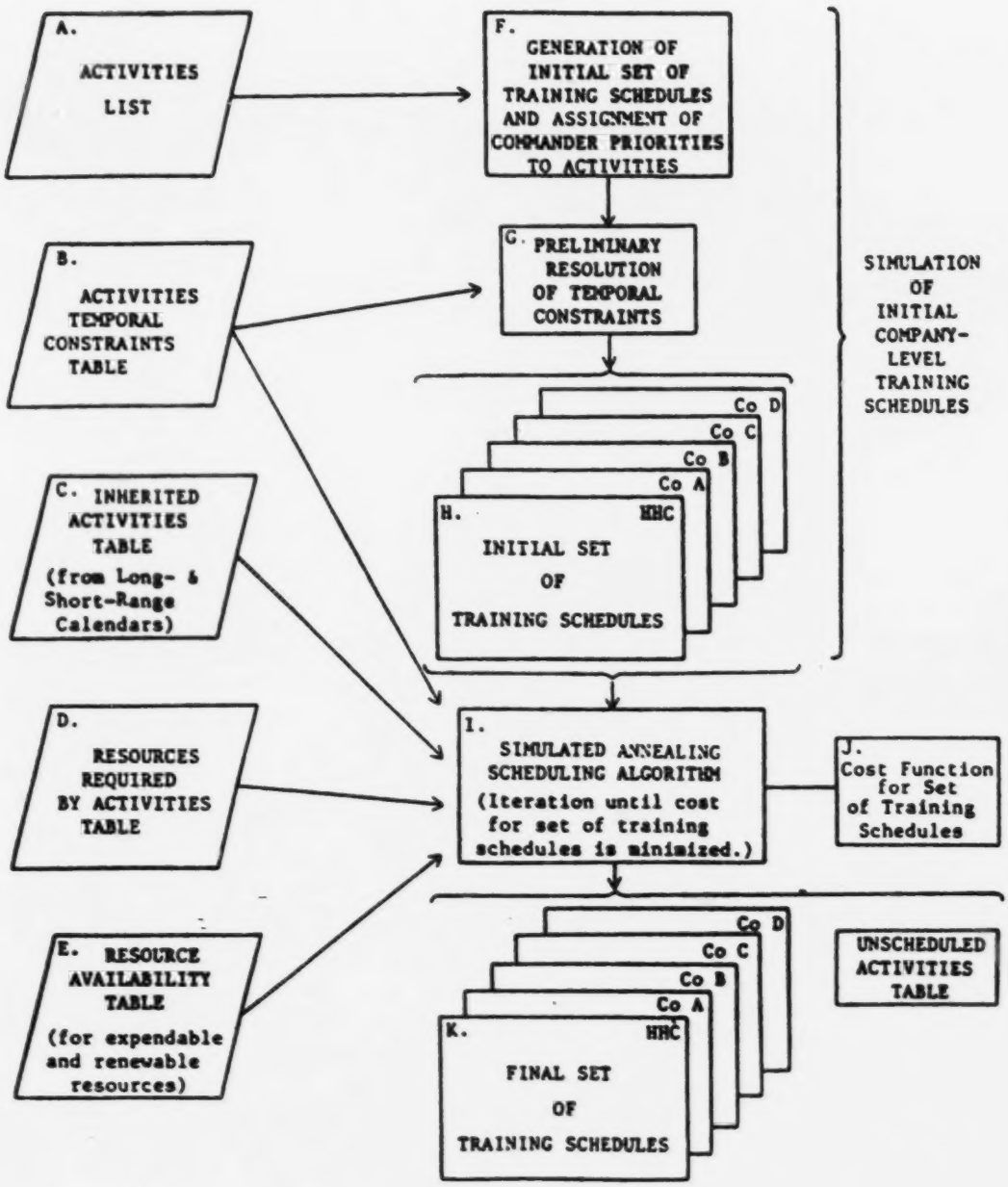


Figure 2. Data Structures and Flow of Control for Simulated Annealing Test Program

It should be emphasized that the input files and output schedules do not represent files as they would be seen by users in a final scheduling program. Users of an implementation version will use input/output presented by a relational DBMS. The DBMS input/output will interface with the FORTRAN code that implements the simulated annealing heuristic.[2]

Initial company-level Training Schedules developed by company commanders are simulated in Blocks F, G, and H of Figure 2. These schedules identify tasks for the week, assign priorities to the tasks ("high" versus "regular" priority, and place activities in temporal order consistent with prerequisite relationships (before/after). These initial schedules represent recommendations from a lower echelon for training schedules that are passed upward (See Figure 1). At battalion level, resourcing is accomplished and conflicts are resolved. As depicted in Figure 2, this conflict resolution is accomplished by the passage of the initially recommended schedules to the simulated annealing algorithm in Block I. The simulated annealing algorithm produces the final set of training schedules for the companies within a battalion. In addition, lists of activities from the initially recommended schedules that could not be accommodated in the final schedules are given along with their level of importance as defined by their contribution to the cost function.

The cost function for the test program was formulated on the assumption that it is costly to fail to schedule activities. Further, it is more costly to fail to schedule longer than shorter activities and more important than less important activities.

More Specifically, let C represent cost and K represent weights reflecting the relative importance of unscheduled activities. Let U represent the time associated with unscheduled activities. Let i represent the activity number among m unscheduled activities within a Training Schedule, and let j represent the Training Schedule number that varies from 1 to n, the number of Training Schedules. Since there is a Training Schedule for each company, there are five Training Schedules per battalion ($n = 5$). Given these definitions, the cost function can be written by Equation 1 as:

---

[2]An important human factors problem was identified in previous automated scheduling research (see Medeiros & Yang, 1983a, 1983b) which will be corrected in future work. Figure 2 depicts five separate input files containing lists of activities and/or resources. These files were separated because they are conceptually distinct, and have parsimonious structures separately. Different files contain common subsets of activities and/or resources. In previous scheduling programs, users were required to enter the same items more than once in separate files. This procedure frequently produced data entry errors, due to heavy recall requirements for users who had to remember previously entered lists, and spelling variations of identical list items. These problems will be removed by having lists stored in some common data base to the extent possible. In addition, lists that are entered would only be typed in once by the user. Input would include automatic retrieval of previously entered lists for subsequent use in the context of different files, avoiding duplicate entry errors.

$$C = \sum_{J=1}^{n} \sum_{i=1}^{m} K_{ij} U_{ij}$$

In order to create the algorithm that "cools" the system, activities need to be partitioned and sequenced for scheduling. In the test program, activities were partitioned and sequenced by the echelon level of the scheduling decision. Since activities appearing on the Long- and Short-Range Calendars are created above the battalion level, they are inherited at the battalion level and are scheduled first. (See Block 2 of Figure 3.). For the test program, it is assumed that resource conflicts and temporal constraints of inherited activities have been previously resolved. Nevertheless, the program checks inherited activities and resolves resource conflicts and temporal precedence of violations if they occur. Inter-schedule conflicts between company-level Training Schedules are resolved next as shown in Block 3 of Figure 3. These conflicts generally involve conflicts over resources. Company-level training priorities (assigned at company level) are scheduled next as shown in Block 4. Then, temporal constraints yielding intra-schedule conflicts are resolved. Finally, any free blocks of time are filled with unconstrained activities yielding an end set of Training Schedules. This decision-making sequence is an oversimplification of the real scheduling process in some respects, excluding some details, but it is sufficiently accurate and complete for evaluation of a test program.

The heating and cooling process occurs iteratively for all company-level Training Schedules in a battalion, in search of a battalion set of Training Schedules with a low overall cost function. The weights assigned to the cost function (K_i), that determine the importance of unscheduled activities, correspond in order to the "cooling" sequence just outlined.

Evaluation of Test Program. To evaluate the simulated annealing test program, hypothetical scheduling problems were devised by creating the five required data entry tables shown in Figure 2 (labelled A-E). The "easy" scheduling problem had (a) an average of four activities per company inherited from the higher echelon calendars, (b) a total of ten activities (from the 60 possible) which required one or more of four different resources, (c) ten Percent of the initially scheduled activities designated as company-level commander training priorities, and (d) five activities with some type of temporal relationship to another activity. The "difficult" scheduling problem had: (a) an average of 12 inherited activities per company, (b) 30 activities requiring one or more of 12 types of resources, (c) 20 percent of initially scheduled activities designated as priorities, and (d) 15 activities in temporal relationships with other activities. Final sets of schedules for the five companies of a battalion produced by the test program for these problems appear to be satisfactory in quality and face validity.

Figure 4 presents the aggregation of ten initial schedule sets of an "easy" and another of a "difficult" scheduling problem. Each type of problem was run for 100 iterations. The cost values, which were larger in an absolute sense for the difficult problem, have for both problems been mapped onto the zero to one interval for comparison. As the improvement in schedule set

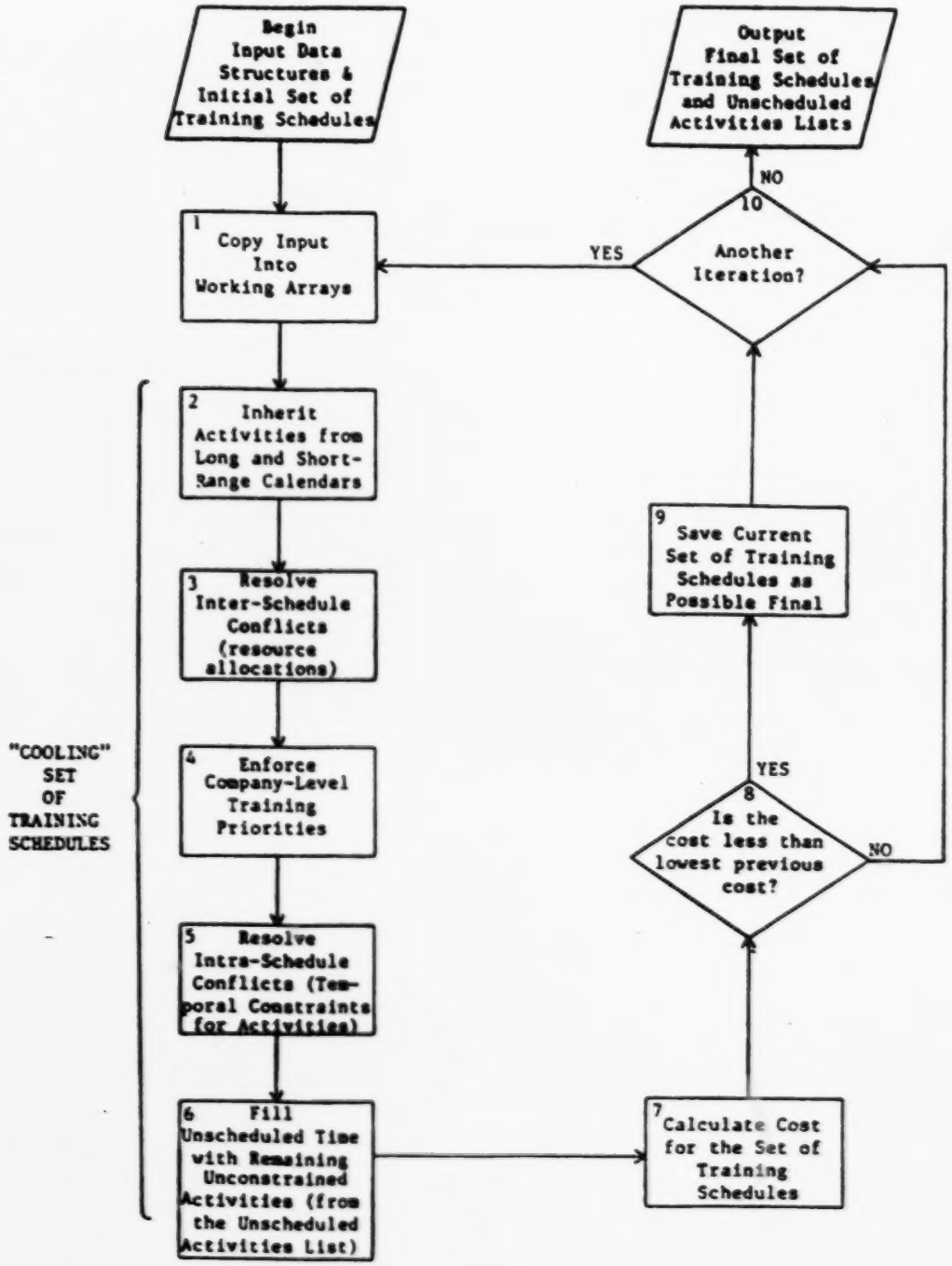

Begin
Input Data Structures & Initial Set of Training Schedules
1 Copy Input Into Working Arrays
2 Inherit Activities from Long and Short-Range Calendars
3 Resolve Inter-Schedule Conflicts (resource allocations)
4 Enforce Company-Level Training Priorities
5 Resolve Intra-Schedule Conflicts (Temporal Constraints for Activities)
6 Fill Unscheduled Time with Remaining Unconstrained Activities (from the Unscheduled Activities List)
"COOLING" SET OF TRAINING SCHEDULES
7 Calculate Cost for the Set of Training Schedules
8 Is the cost less than lowest previous cost?
YES
NO
9 Save Current Set of Training Schedules as Possible Final
10 Another Iteration?
YES
NO
Output Final Set of Training Schedules and Unscheduled Activities Lists

Figure 3. Operation of the Simulated Annealing Scheduling Algorithm

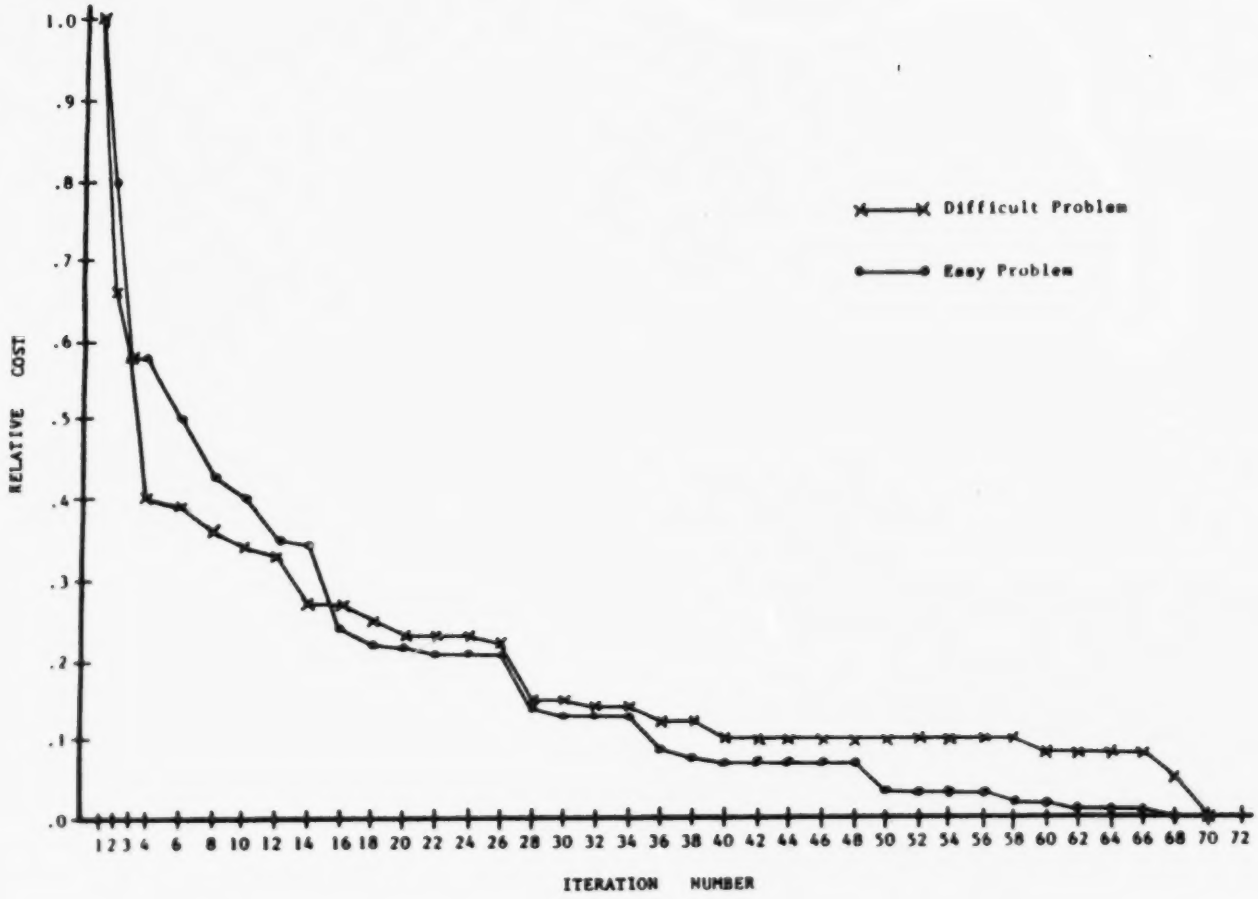


Figure 4. Relative cost by iteration for best schedule sets using easy and difficult scheduling problems.

quality was negligible after the seventieth iteration, the graph is truncated.

An important finding shown in Figure 4 is the rapid rate of schedule set improvement, as illustrated by the rapid decline of the cost function. The improvement in the cost value reaches its asymptotic limit quite quickly, well before 100 iterations. This trend suggests that the current implementation is relatively efficient, at least as indicated by the number of iterations necessary to achieve a stable solution. A second finding is the very similar rate of decrease of the cost function for both easy and hard problems.

Run-time statistics were also examined to further investigate the performance characteristics of the test program. Running on a VAX 11/780 for 100 iterations, the easy problem required an average of about four minutes of CPU time. The difficult problem required about 5.7 minutes of CPU time. Wall clock time was about ten percent more when saving only the best schedule set at any given point. These problems used a maximum of 98,000 bytes of memory, including data tables.

Given above statistics, it may be feasible to run scheduling problems of this size on microcomputers. However, "overnight" runs would probably be required. On one hand, an operational program could make more efficient use of computer resources than the test program. In addition, it is unlikely that real problems would require 100 iterations. On the other hand, actual Army scheduling in practice may produce much larger data tables which could be so voluminous that it might be impossible to enter all data in the memory of a microcomputer at one time. This situation would slow the program operation down considerably.

Operation of the Test Program. Appendix B shows a sample run with complete training schedules for each of five companies at each successive stage of one iteration for the difficult problem. (The five data tables defining the difficult scheduling problem are also presented). To illustrate the types of schedule manipulations the system produces, Table 1 shows a segment extracted for Company B on Monday at each of the six stages in one "cooling" iteration.

The first column of Table 1 shows the initial schedule for Company B produced by randomly sampling from the population of 60 activities without replacement for each of the five companies. Three activities were randomly selected as company-level training priorities and are marked with an asterisk. This initial schedule can be considered the company-level proposed training schedule that is forwarded to battalion for verification and approval.

The second column shows the schedule segment after activities have been inherited from the higher echelon calendars. As shown in Table 1, two activities appear at the required hours designated by the Long- and Short-term schedules: PT and its necessary immediate successor, PERS HYGIENE. Other activities from the initial schedule are inserted either at their designated times, or, if times are not specified, at randomly selected times. (No activities with unspecified times occurred in this schedule segment). All inherited activities are fixed in the schedule for the duration of the iteration. Activities with designated times will not be moved but may be

Table 1
Illustration of Schedular Functioning
for Monday, Company B (from Appendix B)

	Stage of Algorithm					
Time	Initial Schedule	Inheritance	Resource Allocation	Company-level Priorities	Temporal Relationships	Final Schedule
8:00	*MISSION SUPP	PT	PT	PT	PT	PT
9:00	*ARCRFT REC 1	PERS HYGIENE	PERS HYGIENE	PERS HYGIENE	PERS HYGIENE	PERS HYGIENE
10:00	FIRST AID 4	FIRST AID 4	FIRST AID 4	FIRST AID 4	FIRST AID 4	FIRST AID 4
11:00	INSPECTION 2	INSPECTION 2	INSPECTION 2	INSPECTION 2	NERVE AGNT 1	NERVE AGNT 1
13:00	INSPECTION 3	INSPECTION 3	INSPECTION 3	INSPECTION 3	INSPECTION 3	INSPECTION 3
14:00	*NERVE AGNT 3	NERVE AGNT 3	NERVE AGNT 3	NERVE AGNT 3	NERVE AGNT 3	NERVE AGNT 3
15:00	ID TERR FEAT	ID TERR FEAT	AREA MAINT 4	MISSION SUPP	MISSION SUPP	MISSION SUPP
16:00	INSTALL M18	INSTALL M18	PREP M72 1	PREP M72 1	MAINT M16 1	MAINT M16 1

*Activities designated as company-level training priorities

removed from the schedule due to resource or temporal constraint violations. Those with randomly chosen times may be moved or removed. The activities that are removed are placed on the initial unscheduled activities list for the company being processed. In this segment, MISSION SUPPORT and ARCRFT REC 1 went on the unscheduled activity list for Company B.

The third column of Table 1 shows the results of the resource allocation step. FIRST AID 4 requires resource INST MOE which is available at the time the activity has been scheduled. ID TERR FEAT needs resource TR ARA 2 which is not available at 1500. In fact, it was initially available at this time, but, (as shown in Appendix B), this resource was allocated at 1500 to Company A, which has a higher unit priority. Therefore, ID TERR FEAT is exchanged for AREA MAINT 4, an activity which was not previously designated time and for which the needed resource is available. Similarly, activity INSTALL M18 requires TR ARA 3 which is unavailable on Monday and the activity is moved to a time on another schedule (not shown) when the resource is available. If a required resource were unavailable at any time, the activity would be added to the unscheduled activity table. When resource allocations have been completed, all activities requiring explicitly allocated resources become fixed in the schedule, although they may still be removed for temporal constraint violations.

At the fourth step, activities designated as company-level priorities are fixed in the schedule. If priority activities have been previously scheduled (such as NERVE AGNT3 in Table 1), they are fixed in the schedule. If priority activities are not currently scheduled, they are found on the unscheduled activities list. These activities are then, if possible, substituted for activities that have not yet been fixed in the schedule. In the current implementation, only unscheduled priority activities not requiring resources are rescheduled. These priority activities are then fixed so that they cannot be removed during this iteration. As shown in column four of Table 1, MISSION SUPP, which had been removed earlier, is rescheduled due to commander priority. The priority activity ARCRAFT REC 1 is rescheduled on Friday (see Appendix B). Activities that are displaced (in this case, AREA MAINT 4) are added to the unscheduled activities list.

In step 5, temporal relationships are resolved. PERS HYGIENE which was placed in the schedule at the inheritance step is verified as an immediate successor to PT. If it had not been present, an attempt would have been made to add it. The activity MAINT M16 2 which appears later on Tuesday in the Company B schedule requires the prerequisite MAINT M16 1. The time slot with an unfixed activity is on Monday at 1600 so the prerequisite MAINT M16 1 is inserted. Again, the displaced activity, as well as any activities with temporal relationships that cannot be satisfied, are added to the unscheduled activities table. Finally, if any unscheduled blocks of time exist in the schedule, activities would be selected from the unscheduled activities table and put into the schedule. No such change occurred in the particular schedule segment in Table 1. Thus, final schedule shown in column 6 of Table 1, is identical to column 5.

After the final schedule has been determined, the cost is computed. As described earlier, the cost is based on the importance and duration of the unscheduled activities. If an activity can be assigned more than one level of importance, due to having more than one type of constraint, the highest

importance weight is assigned to the activity. Costs are computed for each unscheduled activity and summed across unscheduled activities for a company. Then company costs are summed across all companies in a battalion, yielding a total cost value for the schedule set (see Equation 1 and Subroutine Cost in Appendix A).

In an application version of such a program, normally only the final schedule set would be of interest and would be printed. The application programs would also provide the unscheduled activities list, along with the specific reasons for exclusion, so that manual modifications of the schedule could be made by users as they deemed necessary. The purpose here has been to provide insight to the reader of the logic and inner workings of the current developmental scheduling system.

CONCLUSIONS AND RECOMMENDATIONS

The simulated annealing heuristic approach to the Army scheduling problem, as implemented here, appears promising in several respects. First, it seems to effectively accommodate the highly heterogeneous and unique aspects of the problem. The ease with which the approach fits into the hierarchical organizational structure of the Army suggests that the same principles, and perhaps actual sections of code, could be adapted to scheduling of Long- and Short-Range Calendars at brigade or division level. In addition, difficult scheduling problems appear to be handled rather efficiently with minimal increases in computer resource requirements. As such, we believe that the general approach is worthy of continued development.

On the other hand, the existing program has serious limitations when viewed from the perspective of what will ultimately be required in an applications program. The test program incorporated simplifying assumptions such as using hours as the smallest time unit, scheduling one week at a time, and tacitly assuming that a company has only one activity at any given time. Further, certain constraints were imposed upon the nature of the input data. For example, activities on the right side of a temporal relationship (i.e., constrained to occur after another activity) were assumed not to require resources or themselves to appear on the left side of a temporal relationship (i.e., constrained to occur before another activity). None of these simplifications appear to be unchangeable. Rather, they have been, to date, merely temporary limitations that were subordinate to the goal of testing the overall feasibility of the method. Modifications to incorporate additional complexities entail both development time and probably some code expansion. A potential limitation of the approach is that the final schedule set produced typically will not be optimal in a global sense. The cost function goes down relatively quickly, but the "good" schedules as defined by a low cost function, may still contain clearly non-optimal assignments of activities.

Given that the general method of simulated annealing shows promise for solving or at least contributing to a solution to the Army's unique scheduling problems, several development directions for improving schedules have become apparent to us as a result of the work to date. One such modification which appears likely to yield improvements in schedule quality involves allowing variation in annealing schedules. This variation would permit local small-scale heating and cooling. For example, in the current version of the test program, once resources have been allocated, no further resource allocation

occurs for the duration of the iteration. If local variation in the annealing schedules were permitted, it would be possible to reassign resources after temporal constraints are met in a local iterative loop. It may also be productive to explore other ways of having the successive iterations build directly upon the output of the previous iterations rather than beginning completely anew each time, as is now the case.

Another possibility for improving performance might involve reconceptualizing the concept of "temperature." Instead of defining temperature based on stages involving different types of constraints, as is now the case, temperature could be defined based on of the temperature of activities. Some activities could be considered "hotter" than others based on the number of constraints imposed on them. Thus, activities with a greater number of constraints, have a larger number of potential constraint violations and would be scheduled first. Such a redefinition of temperature would constitute a radical modification of the existing system but appears to be worth pursuing.

Another area for future development that would support the software development effort involves evalution of Army Training Schedules. This effort might further explore what constitutes a realistic Army Training Schedule. The definitions of easy and difficult problems used here are useful for exploring the strengths and weaknesses of the simulated annealing method. However, they are somewhat arbitrary in terms of the specifics of actual Army scheduling. There is a need for evaluation of the quality of schedule set output based on its utility to those charged with the actual planning of training. It is important to determine whether the scheduling outputs from the test program are satisfactory and whether they provide sufficient benefits to overcome the costs incurred in their generation.

REFERENCES

Army Development and Employment Agency (June 1985). Functional description for the Integrated Training Management System (ITMS) (Vol. I). Ft Lewis, WA: BDM Corporation.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983). Optimization by simulated annealing. Science, 220, 671-680.

Medeiros, D. J. & Yang, T. (June 1983a). Base scheduling and resource estimation model (Contract No. DABT60-80-C-0049). University Park, Penn: Pennsylvania State University.

Medeiros, D. J. & Yang, T. (December 1983b). Base scheduling and resource estimation model (Contract No. DABT60-80-C-0049). University Park, Penn: Pennsylvania State University.

Van der Eijk, J. A., Ignizio, J. P., & Yang, T. (1981). Base scheduling and resource estimation for the Army Battalion Training Model (Vol. 3), (Contract No. DABT60-80-C-0049). University Park, Penn.: Pennsylvania State University.

Yang, T. & Ignizio, J. P. (1982). Base scheduling and resource estimation for the Army Battalion Training Model (Contract No. DABT60-80-C-0049). University Park, Penn: Pennsylvania State University.